**Science Olympiad Scoring System**

**Software Requirements Specification**

**1. Introduction**

**1.1 Purpose**

This document describes the software requirements for a Science Olympiad scoring system (**SOSS**) at NDSU. This specification is intended for the developers of the system, as well as the people who maintain it.

**1.2 Scope**

The function of the SOSS is to store data for Science Olympiad competitions that NDSU organizes. The ultimate goal of the system is to provide a central location for Facilitators to view and edit scores for each event of the competition. A similar system already exists, but the SOSS shall contain usability improvements, among others.

**1.3 Overview**

The remainder of this document is organized as follows: There will be some definitions of important terms in the next subsection. Section 2 contains a general description of the SOSS. Chapter 3 identifies the specific functional and performance requirements of the system. Chapter 4 explains what work has been completed, what work has been partially completed, and what work still needs to be started.

**1.4 Definitions**

The following definitions may be helpful when discussing the various moving parts of the SOSS:

* **Competition**

The entire Science Olympiad meet that NDSU organizes. A competition consists of many different schools competing in various events against each other.

* **Team**

A group of students from the same school competing in an event. A school may bring multiple teams to a Science Olympiad competition.

* **Event**

An individual contest within a competition consisting of several teams. There are multiple events in a Science Olympiad competition.

* **Facilitator**

The person who officiates an event in the competition. Each event needs to be scored by a Facilitator. They accomplish this by ranking the teams that participated in the order that they placed. A Facilitator may be responsible for multiple events in a competition.

* **Head Facilitator**

The Facilitator that sets up and oversees the competition. This person needs to be able to edit all the data within a competition, as well as assigning Facilitators to events, and viewing all the scores.

* **Division**

Teams and events are grouped into one of two possible divisions: Junior (B) or Senior (C). Teams only participate in events that are specified for their division, and they only compete against other teams within their division.

* **Class**

Each school participating in the competition may be either a Class A school or a Class B school. Teams belonging to schools of different classes may compete against each other, this is simply additional information that needs to be stored for each team.

**2. General Description**

**2.1 Overview**

To give a short overview of the functionality of the SOSS, the following user scenarios are provided from two different points of view: Facilitator and Head Facilitator.

* **Facilitator**

When a Facilitator is ready to input the scores for their event:

1. The Facilitator logs in to the SOSS using their unique username and password.
2. The Facilitator clicks on the event they are scoring and is brought to a form containing the teams participating in that event.
3. The Facilitator inputs their rank for each team participating in the event.
4. A visual queue shows which ranks still need to be input, ensuring that the Facilitator gives every team a score and there are no duplicate scores.
5. The Facilitator submits their scores for their event, and logs out of the SOSS.

* **Head Facilitator**

When the head Facilitator wants to set up a new competition:

1. The head Facilitator logs in to the SOSS using their unique username and password.
2. The head Facilitator specifies a competition name and creates a new competition.
3. The head Facilitator adds new teams and events for the new competition, maintaining separate lists for each division.
4. The head Facilitator creates new Facilitators for the competition, and assigns them to different events

When the head Facilitator wants to view and edit teams and events in a specific division:

1. The head Facilitator logs in to the SOSS using their unique username and password.
2. The head Facilitator chooses an existing competition to view data from.
3. The head Facilitator chooses a division within that competition.
4. The head Facilitator is able to view and edit the teams and events in that division.

When the head Facilitator wants to check the progress of events being scored:

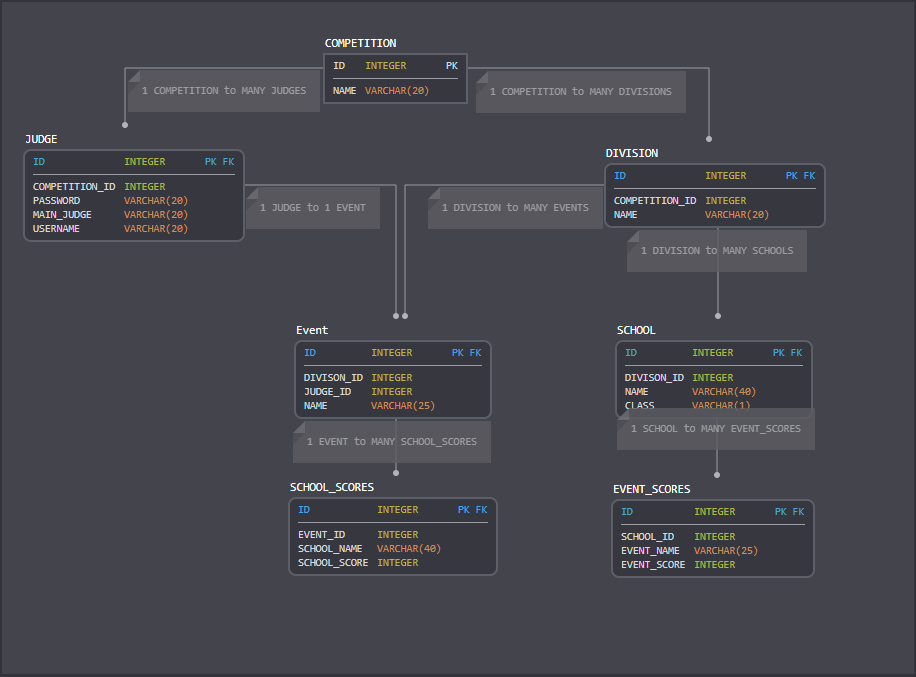
1. The head Facilitator logs in to the SOSS using their unique username and password.
2. The head Facilitator chooses an existing competition to view data from.
3. The head Facilitator chooses a division within that competition.
4. The head Facilitator is able to see at a glance which events have been scored, which events still need scoring, and the percentage of scored events.

When the head Facilitator wants to print a report of the results of a competition:

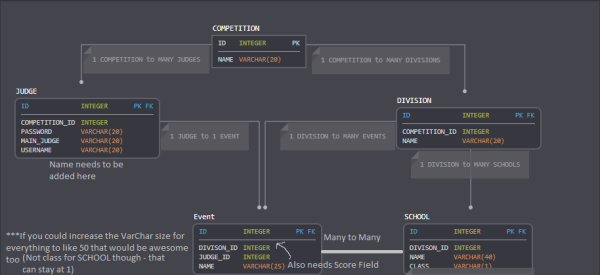
1. The head Facilitator logs in to the SOSS using their unique username and password.
2. The head Facilitator chooses an existing competition to view data from.
3. The head Facilitator chooses a division within that competition.
4. The head Facilitator navigates to the page they wish to print a report from, and selects the option to generate a report.
5. The SOSS generates a report of the data on that page, and downloads it for the head Facilitator to print.

**2.2 Data Perspective**

The following is the initial database schema for the SOSS:



Edited database schema featuring a many-to-many Scores table that connects the Event and School table:



The database would be further expanded upon to include a many to many relationship between the Judge and Event table. The join table is called Event\_Roster and is used to store the related Event and Judge object IDs.

**2.3 User Characteristics**

The users (Facilitators) of the SOSS should require minimal training to navigate through the system.

**2.4 Assumptions and Dependencies**

1. Before a competition takes place, the head Facilitator will create a new competition, add new teams and events for that competition, and create and assign new facilitators to events.
2. A Facilitator can be responsible for multiple events in a competition.

**3. Requirements**

**3.1 Functional Requirements**

This is a list of the functional requirements the SOSS should satisfy. Each grouping of requirements represents a different functionality of the system. They are presented in the following way:

* *Description:*

A description of the specific requirement.

* *Input:*

A description of the inputs that the system gets.

* *Processing:*

A description of what the system should do with the input.

* *Output:*

A description of the response of the system.

**Note:** The input, processing, and output sections are only specified when needed.

**3.1.1 General Requirements**

**Functional Requirement 1**

* *Description:*

The SOSS should store all data (competitions, events, schools, facilitators) for Science Olympiad competitions at NDSU.

**Functional Requirement 2**

* *Description:*

The SOSS needs to have two different interfaces:

1. **Facilitator**: view teams and events, edit scores for the event they are scoring
2. **Head Facilitator**: create new competitions, add and edit teams and events, track progress of scored events, generate reports of the results of the competition

**Functional Requirement 3**

* *Description:*

When a Facilitator who is not the head Facilitator logs into the SOSS, they should only have access to view teams and events in the competition, and edit scores for their event.

* *Input:*

The Facilitator’s username and password.

* *Processing:*

Checks if credentials are valid for a Facilitator who is not the head Facilitator.

* *Output:*

Access to view teams and events, and edit scores for their event.

**Functional Requirement 4**

* *Description:*

When the head Facilitator logs into the SOSS, they should have access to create new competitions, add and edit teams and events, track progress of scored events, and generate reports of the results of the competition, as well as view teams and events.

* *Input:*

The head Facilitator’s username and password.

* *Processing:*

Checks if credentials are valid for the head Facilitator.

* *Output:*

Access to create new competitions, add and edit teams and events, track progress of scored events, and generate reports of the results of the competition.

**Functional Requirement 5**

* *Description:*

The SOSS should only allow access to users with valid usernames and passwords.

**3.1.2 Viewing Requirements**

**Functional Requirement 6**

* *Description:*

The head Facilitator should be able to select a competition and view all data from it.

* *Input:*

The selected existing competition.

* *Processing:*

Fetch all data related to that specific competition.

* *Output:*

Loads pages with teams, events, and any scores for that competition.

**Functional Requirement 7**

* *Description:*

All users should be able to select a division and view all data from that division.

* *Input:*

The selected division.

* *Processing:*

Fetch all data related to that specific division.

* *Output:*

Loads pages with teams, events, and any scores in that specific division.

**Functional Requirement 8**

* *Description:*

Teams should be presented in table form, and all users should be able to see each team’s name, division, class, a list of events the team is participating in, and any scores they have received for those events.

**Functional Requirement 9**

* *Description:*

Events should be presented in table form, and all users should be able to see each event’s name, division, the Facilitator assigned to it, a list of teams that are participating in that event, and the scores input by the Facilitator officiating the event.

**3.1.3 Editing Requirements**

**Functional Requirement 10**

* *Description:*

Only the head Facilitator should have access to add or edit any teams or events.

**Functional Requirement 11**

* *Description:*

When viewing teams, the head Facilitator should be able to edit any team’s name, division, class, or events the team is participating in.

**Functional Requirement 12**

* *Description:*

When viewing events, the head Facilitator should be able to edit any event’s name, division, the Facilitator assigned to it, or teams participating in that event.

**Functional Requirement 13**

* *Description:*

The head Facilitator should be able to add new teams and events for a division within a competition.

**3.1.4 Scoring Requirements**

**Functional Requirement 14**

* *Description:*

A Facilitator should only be able to edit scores for the specific event they have been assigned to.

**Functional Requirement 15**

* *Description:*

The form used for inputting an event’s scores should list all the teams participating in that event, as well as a text box for each team to input a score.

**Functional Requirement 16**

* *Description:*

There should be a visual queue consisting of numbered boxes that change color when a score with that box’s corresponding number is input by the Facilitator.

* *Input:*

A score for a team.

* *Processing:*

Match that number up with its corresponding numbered box.

* *Output:*

Changes that score’s box to green, indicating the Facilitator has input that score.

**Functional Requirement 17**

* *Description:*

The form used for inputting an event’s scores should not allow the Facilitator to submit if not all teams have been assigned a score.

**Functional Requirement 18**

* *Description:*

The form used for inputting an event’s scores should not allow the Facilitator to submit if any score is not a valid integer.

**Functional Requirement 19**

* *Description:*

The form used for inputting an event’s scores should not allow the Facilitator to submit if there is more than one team with the same score.

**Functional Requirement 20**

* *Description:*

The valid range of acceptable scores should be 1-(however many teams are participating in the event). The form used for inputting an event’s score should not allow the Facilitator to submit if there is a score that is outside this range.

**Functional Requirement 21**

* *Description:*

When viewing all the events within a specific division, the head Facilitator should be able to see the number of events that have already been scored, the number of events that still need to be scored, and the percentage of events that have already been scored.

**3.1.5 Report Requirements**

**Functional Requirement 22**

* *Description:*

For each page or table the head Facilitator is viewing data from, they should be able to click a button that generates a report of that page’s data.

**Functional Requirement 22**

* *Description:*

When viewing all events or teams in a specific division, the report generated should display which teams placed 1st, 2nd, and 3rd in the division, as well as all the other teams, their scores for each event, and their overall cumulative score in the competition.

**Functional Requirement 23**

* *Description:*

When viewing a specific event within a division, the report generated should display which teams placed 1st, 2nd, and 3rd in the event, as well as the rest of the teams that participated in that event and their scores.

**Functional Requirement 24**

* *Description:*

When viewing a specific team within a division, the report generated should display all the events that team participated in, as well as their scores in each event, and their overall cumulative score in the competition.

**Functional Requirement 25**

* *Description:*

When the head Facilitator clicks the button that generates a report, an Excel file containing the report data should automatically download.

* *Input:*

Clicking button that generates report.

* *Processing:*

Generates report with data from the current page of the SOSS.

* *Output:*

Downloads an Excel file containing the report.

**3.2 Performance Requirements**

This is a list of the performance requirements the SOSS should adhere to.

**Performance Requirement 1**

* *Description:*

All data requests from the user should be satisfied within 3 sec.

**Performance Requirement 2**

* *Description:*

When the head Facilitator requests a report to be generated, it should download within 3 sec.

**3.3 Attributes**

**3.3.1 Availability**

* The system should be available for the head Facilitator 24 h/day.
* The system should be available for ordinary Facilitators only during Science Olympiad competitions.

**3.3.2 Security**

* Only authenticated users with valid usernames and passwords should have access to the SOSS.

**3.3.3 Maintainability**

* The system should be easy for other developers to maintain, as this will most likely be passed on to the next Principles of Software Engineering class.

**4. Work Progress**

**FULLY IMPLEMENTED:**

The following requirements have been fully implemented in the project, but heavy testing may still need to be done.

* **Two interfaces – Head Facilitator/Admin and Facilitator that contact the database when logging in.**
* **Head Facilitator can create and edit all data (competitions, facilitators, events, teams) at any time.**

**SOMEWHAT IMPLEMENTED:**

The following requirements have been partially implemented in the project, and a description is included to explain what still needs to be done.

* **Head Facilitator can generate a full report for all teams, events, and their scores in each division, but more still needs to be completed:**
  + Need to calculate the totals for each team in the report.
  + Other reports need to be generated, like for each event and for each team.
* **Head Facilitators can view which events a facilitator is assigned to, but physically assigning an event to a facilitator needs more work:**
  + The process of selecting a facilitator from the table, then choosing an available event from a drop-down list is complete, but the connection when submitting is broken.
* **Facilitator interface will contain a scoring page where they can only score the events they are assigned to:**
  + The pages are there, but the connections and process need to be fixed and improved.

**NOT IMPLEMENTED:**

The following requirements have not been implemented in the project, or they have, but the components have no functionality.

* **Head Facilitator needs to be able to select a competition to view all data from.**
  + Currently just taking the first competition in the table.
* **Head Facilitator interface will contain a visual queue that will display scoring progress and which events still need to be scored.**
* **Full test of completed system with product owner.**